Automatic Composition of Guzheng (Chinese Zither) Music Using Long Short-Term Memory Network (LSTM) and Reinforcement Learning (RL)

Shuling Chen

South China University of Technology

Yong Zhong (✉ zhongyong@scut.edu.cn)

South China University of Technology

Ruxu Du

South China University of Technology

Research Article

Keywords: Music composition, Guzheng (Chinese Zither), Long Short-Term Memory network (LSTM), Reinforcement Learning (RL)

Posted Date: January 31st, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1268083/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Abstract

In recent years, with the advance of Artificial Intelligence, automatic music composition has been demonstrated. However, there are many music genres and music instruments. For a same piece of music, different music instruments would produce different effects. Invented some 2500 years ago, Guzheng is one of the oldest music instruments in China and the world. It has distinct timbres and patterns that cannot be duplicated by other music instruments. Therefore, it is interesting to know whether AI can compose Guzheng music or alike.

In this paper we present a method that can automatically compose and play Guzheng music. First, we collect a number of existing Guzheng music pieces and convert them into Music Instrument Digital Interface (MIDI) format. Second, we use these data to train a Long Short-Term Memory (LSTM) network and use the trained network to generate new Guzheng music pieces. Next, we use the Reinforcement Learning (RL) to optimize the LSTM network by adding special Guzheng playing techniques and Guzheng timbre. According to the evaluations from general audiences, our Guzheng music is very close to the real Guzheng music. The presented method can also be used to automatically compose the music of other Chinese music instruments.

1. Introduction

Music is a universal art enjoyed by all the people around the world. It is well known that music has different genres and can be played using different music instruments. Music composition has always been considered as creative art that only talented musicians can master. In recent years, the technology of Artificial Intelligence (AI) has been greatly advanced. There have been a number of successful attempts to use AI for music composition. However, the success is rather limited as only a few kinds of music, such as piano music and guitar music, can be generated. Particularly, there has been no reported success in composing traditional Chinese music.

Among various traditional Chinese musical instruments, Guzheng (Chinese Zither)(古筝) is perhaps ranked at the top in popularity. Invented some 2500 years ago [1], through years of inheritance and innovation, development and improvement, it has its unique music language, playing style, as well as special identity unlike any others. In fact, just hearing Guzheng music, many people around the world would think of China.

Figure 1 shows the picture of a classical Guzheng. Its body is a hollow box made of sycamore wood. Its strings are placed on the top face while the sound holes are at the bottom (not shown in the figure). From the structure point of view, Guzheng consists of 5 parts: (1) the head of the Guzheng and tuning box, (2) Yue Shan (雁阵) on which the strings are mounted, (3) the column of Guzheng that acted as the bridge for the strings, (4) the panel, on which the strings are crossed, and (5) the tail of Guzheng. According to the historical records, the original Guzheng had only 5 strings. In Three-Kingdom period (220 ~ 280), it had 12 strings. In Tang Dynasty (618 ~ 907), it added to 13 strings, and in Ming Dynasty (1368 ~ 1644) and

Qing Dynasty (1636 ~ 1912) it had 16 strings. In the modern era, it was sublimated to modern musical instrument making method and gradually increased to 18 strings, 21 strings and even 25 strings. Presently, the design of Guzheng is standardized with 21 strings and a fixed length of 1.63 meters long. Its tone is in pentatonic scale with a range of three octaves. The mode is mostly G or D. Also, players need to wear tortoise's bill armor pieces (i.e. false fingernail) to play.

There are a number of special skills for playing Guzheng. The right hand uses the thumb, index finger, middle finger and ring finger to pluck the strings. The fingering methods include Gou (勾), Tuo (挑), Pi (劈) and etc. The left hand presses the strings and the fingering techniques include Press (按), Slide (滑), Chatter (颤) and so on. In recent years, new Guzheng playing techniques are also added such as playing the left and right hand alternatively and playing multi-part together. It creates very rich and vibrant music [2, 3, 4].

There are at least 200,000 people are playing Guzheng with different skill levels. In 2019, there were over 17,000 people participating the national exam, making it the most popular traditional Chinese music instrument. Indeed, many people view Guzheng as important inheritance and have special cultural value. In recent years, although the Guzheng music composition spreads, few music pieces could not reach the height of the traditional Guzheng music pieces. Therefore, it is wondrous if we can use AI to compose good Guzheng music.

AI based music composition, also referred to as computer music generation, refers to generate a specific sequence of music notes with minimum human intervention. Based on literature survey, the first attempt of computer music generation can be dated back to 1950s [5]. The early pioneers proposed to use Markov model to generate music, but the experiments showed that only fragment musical notes were generated. Later research found that the Markov chain model offers only a stitching process and cannot be called music composition. Consequently, Hidden Markov Model (HMM) were proposed which was able to generate polyphony music for the first time [6]. Several efforts followed demonstrating the feasibility of computer music generation [7, 8]. However, HMM has some limitations, such as its state evolution is linear, it is difficult to learn long-term dependence, and it is incapable of dealing with large data sets.

Neural network is another commonly used method for computer music generation [9, 10]. For example, in [11], Schuster and Paliwal used bidirectional Recurrent Neural Network (RNN) to generate computer music. However, because of the gradient disappearance of RNN, it is difficult to learn the long-term correlation within a music piece, resulting in fragmented note cycles. In [12], Eck and Schmidhuber used Long Short-Term Memory network (LSTM) to generate more complex music. LSTM can memorize long-term information and avoid the aforementioned gradient disappearance problem. Consequently, it makes the generated music pieces more coherent. In [13], a new LSTM with Gated Recurrent Unit (GRU) was used obtaining improved results. Subsequently, LSTM is widely used in computer music generation [14, 15, 16]. In [17], in order to solve the problem of melodic level of music generated by LSTM, an algorithm called Music VAE (Variational Autoencoder) is proposed. In [18], VAE is used to improve the quality of audio synthesis. The emerging of Generative Adversarial Networks (GAN) algorithm [19] provides a new method for AI based music composition. GAN consists of two networks: the generating network G

(Generator) and the discriminant network D (Discriminator), the training achieves a Nash equilibrium, making G capable of generating data with the same distribution as that of the training data and D capable of detecting false data, which is ideal for music composing. The use of GAN for music composition is reported in [20, 21], though, it seems the generated music pieces are still full of random notes.

Despite of ever increased efforts in AI based music composition, no one has tried to use computer to generate Guzheng music. This may be attributed to several challenges. Firstly, it is known that the modern digital music format is MIDI. Currently, the MIDI library contains the timbres of many music instruments, such as piano, violin, guitar etc., but not Guzheng. Consequently, it becomes difficult to generate Guzheng music. One may of course use MP3 format, though, it requires much more training samples and computation loads. Yet, it cannot guarantee the success. Secondly, Guzheng has many unique fingering skills, which cannot be readily expressed in MIDI format. Finally, there are few people who know both AI and Guzheng.

In this paper, we present a new method for composing Guzheng music using LSTM and Reinforcement Learning (RL). It consists of three steps: preprocessing, training and music generation, as well as post processing. Accordingly, the rest of the paper is organized as follows. Section 2 describes the pre-processing steps including a brief review of music, the conversion of the Guzheng music to MIDI format, as well as the extraction of chorus. Section 3 discusses the training of LSTM and Reinforcement Learning (RL). Section 4 shows the post-processing step. A set of experiment results are also given. Finally, Section 5 contains conclusions and future work.

2. Pre-processing

2.1 A brief review on the elements of music

There are several different views on the elements of music. From the acoustic point of view, music is made of sounds which can be characterized by its time and frequency characteristics. From the music point of view, though, the definitions vary. The commonly accepted definition is that the basic elements of music include: pitch, duration, loudness, timbre, sonic texture and spatial location. Though, the most important elements of music are timbre and melody.

Timbre is the perceived sound quality of a musical note, a sound or a tone. It distinguishes a sound from the others. Timbre can be characterized by the following attributes: the tone (the arrangement of pitch and/or chord), the spectral envelop, the time envelop, the changes of the frequency and spectral envelop, as well as the onset of the sound [22]. Guzheng, like the other music instruments, has its unique timbre. Fig. 2 shows the sound wave from Piano, Guzheng, Harmonica, and Violin when playing a same piece of song with the same volume. As expected, Guzheng has unique timbre different to all others.

Melody, also called tune, is a timely succession of music tones. It is a combination of pitch and rhythm. Melody often consists of one or more musical phrases and are usually repeated within the music piece.

Guzheng music belongs to traditional Chinese music and has its own melody styles and patterns. In particular, it emphasizes on artistic conception and cavity rhyme. As shown in [23], traditional Chinese music uses five tones: C, D, E, G, and H. Additionally, it usually uses single instrument, single rhythm. Though, modern Chinese music is more diversify and uses the standard seven tones. However, its artistic conception and cavity rhyme are difficult to describe.

In order to generate Guzheng music, it is necessary to sort out the timbre of the Guzheng as well as the melodies of the Guzheng music first.

2.2 Guzheng music data pre-processing

Music data can be represented in two kinds of format: the audio format (MP3 or AVI) and the MIDI format. In this paper, we use the MIDI format as it is concise and easy to edit. MIDI files contain musical notes, timing, and playing definitions of up to 16 channels. For each channel, it includes channel number, length, volume, strength, etc. Because a MIDI file is a series of instructions, it requires little disk space. More importantly, one can easily add, delete or edit the attributes of a note.

We pick Guzheng music pieces from the popular Guzheng repertoires in the Guzheng training textbooks. These music pieces are in the form of numbered notation. Next, we convert these music pieces to form of staff notation. Then, we use MuseScore software to convert them to MIDI format. This process is illustrated in Fig. 3.

A total of 31 pieces of Guzheng music are used for training. These music pieces range from 30 seconds to 4 minutes in length. It shall be pointed out that MIDI has the play definitions of many music instruments, such as piano, violin, guitar and etc., but not Guzheng. From the mechanics point of view, Guzheng is similar to Guitar, which generate sounds with one hand pressing the strings while the other hand toggling the strings. However, Guzheng has 21 strings and can play register from D2 ~ D6 (International phonetic alphabet) while Guitar can only play register from E2 ~ C6. Therefore, we use piano to animate Guzheng, as it can play 88 notes, for music generation. The generated music will be converted to Guzheng music by converting to the Guzheng timbre in post-processing, which will be shown in Section 4.

2.3 Extraction of the melodies

As pointed out in earlier, music is characterized by its melody. Thus, we extract the melodies of the training samples use them to enhance the training. In general, melody may appear in two ways: (1) For simple and short music, its melody is the chorus which replicates itself with minor variations. (2) For complicated and long music, its melody is characterized by its tensed motion and cadence.

For capturing the melodies, we use the similarity matrix. In the MIDI format, the music is scanned at a rate of 10 frames per second, or one frame every 0.1 second. Each frame is represented by a 128-dimensional vector giving 128 pitches in total. The value in each dimension represents the strength of the note in the corresponding pitch. Thus, each frame can be described by a vector, $v = [x_1, x_2, \dots, x_{128}]$. Moreover, a

piece of music can be described by a matrix $X = [v_1, v_2, \dots, v_N]^T$ where, N number of frames. Accordingly, the similarity matrix is a $(N-1) \times (N-1)$ matrix, whose element is as follows:

$$S_{i,j} = \sqrt{(v_i - v_{i+1})^2 + (v_j - v_{j+1})^2}$$

1

where, $i, j = 1, 2, \dots, N-1$.

Figure 4 shows the similarity matrix of a piece of music. In the figure, different levels of brightness represent different levels of similarity. It is seen that the ones marked by yellow boxes have a similar pattern and hence, are correspondent to the melody.

The similarity matrix works fine for finding repeats with minor variations. For repeats with relatively large variations, we can use the similar index defined below:

$$S = 1 - \frac{\|X_{i-12} - X_i\|}{\sqrt{12}}$$

2

where, 12 corresponds to the number of basic notes, X_{i-12} and X_i are the matrices corresponding to the two segments in the music piece. If the value of S is high, then the two segments are similar.

As an example, Fig. 5 illustrates the extraction of a melody.

There are Guzheng music pieces that do not have repetitions, such as “Gao shan liu shui (高山流水)” and “Yu zhou chang wan (渔舟唱晚)”. In this case, we will select the segment with high intense and cadence.

3. Training Using Reinforcement Learning

3.1 A brief introduction of LSTM network

In this paper, we use LSTM network for Guzheng music generation. Fig. 6 shows a LSTM unit. It has three inputs: the data input (note vectors), X_t , as well as the outputs from the previous unit, h_{t-1} and C_{t-1} . Here, h_{t-1} is the output while C_{t-1} represents how well the output shall be retained. As shown in the figure, the LSTM unit is made of three gates: the forgetting gate (in blue), the input gate (in pink) and the output gate (in yellow).

The forgetting gate is used to control how much information shall be “forgot” from the previous unit and is computed as follows:

$$f_t = \delta(W_f \cdot [h_{t-1}, X_t] + b_f)$$

3

where h_{t-1} is the output of the previous unit, X_t is the input, W_f is the weight, b_f is the bias, and δ is the Sigmoid function. $f_t \times C_{t-1}$ determines how much information from the previous

unit will retain.

The input gate is used to determine the effect of the inputs. It contains two parts:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \\ c_t &= \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \end{aligned}$$

4
where W_f and W_i are the weights, b_i and b_c are the biases.

The output gate is used to determine the information output. Just like the previous unit, it gives two outputs: h_t is the output and C_t represents how well h_t shall be retain. The output h_t is computed as follows:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \\ h_t &= o_t \cdot \tanh(W_h \cdot [h_{t-1}, X_t] + b_h) \end{aligned}$$

6
where W_0 is the weight and b_o is the bias. The control C_t is computed as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot c_t$$

7
It is interesting to note that when $f_t \rightarrow 1$ and $i_t \rightarrow 0$, $C_t \approx C_{t-1}$, which retains all the information from the previous unit. On the other hand, when $f_t \rightarrow 0$ and $i_t \rightarrow 1$, $C_t \approx c_t$, which retains no information from the previous unit.

In this paper we use 3 LSTM, 2 Dropouts, 2 Denses and 1 Softmax as shown in Fig. 7. In the setting, the LSTM is described by three parameters: the sequence length, the characteristic number of the hidden layer, and the number of weights in the network. The Dropout is used to reduce overfitting by hiding 30% of the neurons. In the setting, it is described by two parameters: the sequence length, the characteristic number of the hidden layer. In the setting, it can be described by three parameters just like LSTM. The Softmax function is used to generate music notes. Given data sets, $\{h_{j,t}\}, j = 1, 2, \dots, M, t = 1, 2, \dots, N$, it first computes the probability:

$$P(h_{j,t}) = \frac{e^{h_{j,t}}}{\sum_{j=1}^M e^{h_{j,t}}}, j=1, 2, \dots, M, t=1, 2, \dots, N$$

8
Then, the ones with the highest probability is selected as the generated note:

$$y_t = \underset{j}{\text{argmax}} P(h_{j,t}), t=1, 2, \dots, N$$

9
The training of LSTM is based on the minimization of the cross-entropy function:

$$L = -\frac{1}{M} \sum_{j=1}^M (\sum_{t=1}^N (Y_{j,t} \log(o_{j,t}) + (1 - Y_{j,t}) \log(1 - o_{j,t})))$$

where, $\{Y\}_{-j} = \frac{1}{128} \sum_{t=1}^{128} \left(\{X\}_{-j,t} - \{Y\}_{-j,t} \right)^2$.

During the training, we set the learning rate as $\{\alpha\} = 0.4 \times 10^{-3}$. Also, considering the lack of contextual information in the first 6 notes of each musical piece, these notes are not be used.

In the setting, it can be described by the sequence length and the characteristic number of the hidden layer. In summary, Table 1 shows the network parameters.

Table 1
The network parameters

Name	Setting parameters
LSTM1	The sequence length = 100; The characteristic number of the hidden layer = 512; The number of weights in the network = 1,052,672;
Dropout1	The sequence length = 100; The characteristic number of the chidden layer = 512;
LSTM2	The sequence length = 100; The characteristic number of the hidden layer = 512; The number of weights in the network = 2,099,200;
Dropout2	The sequence length = 100; The characteristic number of the hidden layer = 512;
LSTM3	The sequence length = 100; The characteristic number of the hidden layer = 512; The number of weights in the network = 2,099,200;
Dense1	The sequence length = 100; The characteristic number of the hidden layer = 256; The number of weights in the network = 131,328.
Dropout3	The sequence length = 100; The characteristic number of the hidden layer = 256;
Dense2	The sequence length = 100; The characteristic number of the hidden layer = 138; The number of weights in the network = 35,466;
Softmax	The sequence length = 100; The characteristic number of the hidden layer = 138;

3.2 An introduction of Guzheng playing techniques

Guzheng is a two thousand years old music instrument. Throughout the history, many playing techniques were added or modified. Currently, there are many complicated playing techniques, each has its own characteristics. It is impossible to summarize all the playing techniques, as new ones are continuously being invented and old ones are being modified. We therefore focus on the typical playing techniques instead. In general, the playing techniques can be divided into two categories: the left-hand playing techniques and the right-hand playing techniques.

The left-hand playing techniques imply the use of left-hand on the left side of the bridge. It changes the tension of the string, mainly by pressing the string, with the middle finger, ring finger or index finger. Its basic task is to decorate the play of the right hand, so that the tone is richer. Table 2 shows three typical left-hand playing techniques with their playing methods, usual location in a music piece and a sample timbre.

The right-hand playing techniques imply the use of right hand playing. Right hand playing is mainly to pluck the strings, and the position of the playing is mainly on the right side of the bridge. Its basic task is to play the tone. Depending on the music note, the player can use the index finger, the big finger, ring finger and/or middle finger to play. The key is to control the duration, loudness and sonic texture. Also, it is not unusual to play multi-tones at a same time. Table 3 presents four typical right-hand playing techniques with their playing methods, usual location in a music piece and a sample timbre.

3.3 An introduction of Reinforcement Learning

When LSTM network learns the dependence among notes from the dataset, it may or may not catch the characteristics of Guzheng music. Thus, we use Reinforcement Learning (RL) to refine the music note generated from the LSTM network.

The theory of RL has been well developed [24, 25]. Fig. 8 shows the agent-environment interaction in the RL process. At time t , following a policy, π , and the given state, $\{s\}_t$, the agent generates an action, $\{y\}_t$. According to the action, a reward, $\{r\}_{t+1}$, and a new state, $\{s\}_{t+1}$, will then be generated through the environment, which in turn will be input to the agent again in the next round. This process continues until satisfactory result is obtained.

We use Deep Q-learning Network (DQN) as the core algorithm for RL. It is a model-free algorithm based on Q-learning and its goal is to select appropriate policies to optimize the sum of rewards at the end of each epoch. In this algorithm, the agent is a Deep Neural Network (DNN), i.e.:

$$\{y\}_t = \pi(\{s\}_t) = \text{DNN}(\{s\}_t)$$

11

The reward policy is designed to catch the characteristics of Guzheng music. As an example, let us consider Arpeggio (refer to Table 3). It is a technique usually consisting of seven notes. Many even-beat positions in Guzheng music end with arpeggio. Thus, assuming that at time t , the input state is $\{s\}_t = \left\{ \{A\}_t^1, \{A\}_t^2, \{A\}_t^3, \{A\}_t^4, \{A\}_t^5, \{A\}_t^6, \{A\}_t^7 \right\}$, where

$\{A_{-t}^1, A_{-t}^2, \dots, A_{-t}^7\}$ represents the seven notes incur at t , then, the reward policy is defined below:

$r_{-t}^1(s_{-t}, y_{-t}, t) = \begin{cases} 0.5 & \text{if } y_{-t} \in \{A_{-t}^1, A_{-t}^2, \dots, A_{-t}^7\} \text{ and } (t = \text{even beat}) \\ -0.5 & \text{otherwise} \end{cases}$

12

The other example is finger shake technique. We consider that when 4 same notes occur, it is the finger shake. The reward policy, r_{-t}^2 , is to reduce the velocity by half to make it sounds more like finger shake:

$r_{-t}^2(s_{-t}, y_{-t}, t) = \begin{cases} 1 & \text{if } y_{-t} \in \{A_{-t}^1 = A_{-t-1}^1 = A_{-t-2}^1 = A_{-t-3}^1\} \\ -1 & \text{otherwise} \end{cases}$

13

As for big pinch and small pinch, they both contain two notes. Big pinch usually appears in downbeat while small pinch usually appears in upbeat. Thus, the reward policy is

$r_{-t}^3(s_{-t}, y_{-t}, t) = \begin{cases} 1 & \text{if } y_{-t} \in \{A_{-t}^1, A_{-t}^2\} \text{ and } (t = \text{odd beat}) \\ -1 & \text{otherwise} \end{cases}$

14

In general, the total reward is:

$R_{-t} = \sum_{j=1}^n r_{-t}^j(s_{-t}, y_{-t}, t)$

15

where n is the number of Guzheng characteristics. Assuming that the previous total return is:

$R_{-t}^{\text{total}} = R_{-t} + \gamma R_{-t-1} + \gamma^2 R_{-t-2} + \dots + \gamma^{n-t} R_{-1}$

16

where, $\gamma \in [0, 1]$ is a discount value. Then, the average rate of return, $\bar{R}_{-t}$, is:

$\bar{R}_{-t} = \frac{1}{t} \sum_{i=1}^t R_{-i}$

17

Depending on the action, y_{-t} , the value of the action is as follows:

$Q(s_{-t}, y_{-t}; \theta_{-t}) = \underset{y_{-t}}{\text{max}} \{Q(s_{-t}, y_{-t}; \theta_{-t})\}$

18

where, θ_{-t} is the weights of the DNN at time t . By continuously updating the right-hand side of the formula below, we can reach the target Q -value, $\widehat{Q}(s_{-t}, y_{-t}; \theta_{-t})$:

$\widehat{Q}(s_{-t}, y_{-t}; \theta_{-t}) \leftarrow (1 - \alpha) Q(s_{-t}, y_{-t}; \theta_{-t}) + \alpha [R_{-t} + \gamma \underset{y_{-t}}{\text{max}} \{Q(s_{-t+1}, y_{-t}; \theta_{-t})\}]$

19

where, $\alpha \in [0,1]$ is a parameter to control the effect of the previous actions and s_{t+1} is the new state. In this study, we choose $\alpha = 0.15$ and $\gamma = 0.8$.

In each iteration, a data set (s_t, R_t, y_t) is generated and stored in an experience pool, D . The objectives of the RL is to train the DNN to minimize the following loss function:

$$L = \{E\}_D \left[\left((R_t + \gamma \underset{\{y_t\}}{\text{max}} Q(s_t, y_t; \{\theta_t\}) - Q(s_t, y_t; \{\theta_t\}) \right)^2 \right)$$

20

where, $E[\cdot]$ is the mathematical expectation, $\{\theta_t\}$ is the parameter of the target DNN. This optimization can be solved using gradient descent method. It converges when $Q(s_t, y_t; \{\theta_t\})$ reaches $(R_t + \gamma \underset{\{y_t\}}{\text{max}} Q(s_t, y_t; \{\theta_t\}))$.

The process of RL consist of the following steps:

Step 1: Initialize the action value function y_t , initialize the experience pool D , initialize the DNN parameter $\{\theta_i\}$, set the value of the action Q as a random number, and set $\{\theta_i\} = \{\theta_i\}$;

Step 2: Enter the notes y_t generated by LSTM (refer to Eq. (9)) as the initial value of y_t ;

Step 3: Compute the reward, R_t , using Eq. (15) and the average rate of return, $\text{stackrel{-}{R}}_t$, using Eq. (17);

Step 4: Compute the value of action, $Q(s_t, y_t; \{\theta_t\})$, using Eq. (18) and the target Q-value, $\widehat{Q}(s_t, y_t; \{\theta_t\})$, using Eq. (19);

Step 5: Solve the loss function defined in Eq. (20) to get $(s_t, R_t; \{\theta_t\})$, and put it into the experience pool D ;

Step 6: If the target is reached ($\text{stackrel{-}{R}}_t = 1$ and $L \rightarrow 0$), stop; else update $\{\theta_t\} = \{\theta_{t+1}\}$, $t = t + 1$, goto Step 3.

3.4 The whole training process

The training program is written in Python using the following tools in the Python toolkit: TensorFlow V1.5, Keras V2.1.5, Music21 V5.2.0, Torch V1.8.0, and Mido V1.2.0. It consists of 7 steps:

Step 1: Covert the music notes to MIDI format, through which a piece of music can be represented as a matrix: $\mathbf{x} = \{x_{t,j}\}$, $t = 1, 2, \dots, N$, $j = 1, 2, \dots, 128$;

Step 2: Encode the data using One-Hot coding, which gives the input matrix: $\mathbf{X} = \{X_{t,j}\}$, $t = 1, 2, \dots, N$, $j = 1, 2, \dots, 128$;

Step 3: Feed the data to the network described in Fig. 7, which gives the output matrix: $\mathbf{Y} = \{\{Y\}_{t,j}\}$, $t = 1, 2, \dots, M, j = 1, 2, \dots, 128$;

Step 4: Decode the data, which gives $\mathbf{y} = \{\{y\}_{t,j}\}$, $t = 1, 2, \dots, M, j = 1, 2, \dots, 128$;

Step 5: Train the DQN network until convergence;

Step 6: Convert back to MIDI format;

Step 7: Convert to the music note.

4. Post-processing And Experiment Results

4.1 Converting the music with Guzheng timbre

As shown in the previous section, we use LSTM to generate Guzheng music and use RL to enhance the generated Guzheng music. However, the music is in piano timbre. Therefore, a post-processing step is necessary.

It is known that MIDI has many built-in timbre settings for different music instruments and sounds, but not for Guzheng. As discussed in Section 2, we generate the Guzheng music using piano timbre. To convert the piano timbre to Guzheng timbre we used a third-party freeware in Fig. 9 for the conversion (<https://www.cnblogs.com/qingjun1991/p/4971514.html>).

It shall be noted that piano and Guzheng have different range of sound. Fig. 15 shows the correspondents between the piano keyboard and Guzheng strings, marked in yellow. The 21 basic Guzheng sounds corresponds the 21 tones on the piano with the central C key of the piano corresponds to the 12th string of Guzheng. As we use the piano MIDI to generate the music, it is possible that some music notes are out of the range of the Guzheng. In this case, these music notes will be simply deleted.

4.2 Experiment results

Using the aforementioned method, new music pieces are generated. First, we decompose the MIDI files of the 31 Guzheng music into segments. Each segment has 1000 lines, covering 10 sec. Then, we randomly pick these segments as the input of the LSTM. Upon receiving the input, the LSTM will generate an output, also lasts 10 sec. The outputs are evaluated using the loss function defined in Equation (10). The number of inputs is referred to as the Batch size. The larger the Batch size, the better the LSTM can learn, but the more the computation load. Fig. 10 shows the loss function against the batch sizes of 128, 256 and 512. From the figure, it is seen that the larger Batch size require longer time to converge. As an example, Fig. 11 shows a music piece generated using Batch size_of 512. It consists of 4 segments and hence, is 40 seconds long. Its audio is uploaded in the webpage <https://www.bilibili.com/audio/au2743716?type=1>.

The generated music pieces are usually lack of melody and Guzheng characteristics. Thus, as discussed in Section 2.3, we extract the melodies from the training samples pieces and mixed them with the original 31 Guzheng music pieces as the training samples and run the LSTM again. A sample result is shown in Fig. 12. The music audio is uploaded in the webpage: <https://www.bilibili.com/audio/au2744025?type=1>

Additionally, we use RL network to add the Guzheng characteristics. For RL, the outputs are evaluated using the average rate of return defined the average ratio of total revenue to optimal revenue in the current training session and the loss function defined in Equation 16. As shown in Fig. 13, it coverages after 1400 epochs. A sample of generated music is shown in Fig. 14, in which the Guzheng characteristics are marked in blue. The music audio is uploaded in the webpage: <https://www.bilibili.com/audio/au2745842?type=1>.

Figure 15 shows one of the completed Guzheng music piece in which the Guzheng characteristics are marked in pink. The music audio is uploaded on the webpage: <https://www.bilibili.com/audio/au2745999?type=1> .

We generate a number of Guzheng music pieces and ask some 20 volunteers to evaluate. According to their evaluations, our Guzheng music is very close to the real Guzheng music.

5. Conclusions And Future Work

This paper presents a new method for generating Guzheng music using LSTM and RL. Based on the discussions above, following conclusions can be drawn:

- (1) The presented method is effective. According to the evaluation by some 20 people, the generated music pieces resemble the classic Guzheng music very well.
- (2) The key to the success is to extract the melody from the training sample and use LSTM network to train. Additionally, the use of RL adds the Guzheng characteristics to the music making it more like a Guzheng music.
- (3) The presented method is efficient. Using the MIDI format, the training and music generation take only several hours in a PC computer.

We expect that the same method can be applied to generate music for other traditional Chinese music instruments, such as Pipa (琵琶), Guqin (古琴), Lusheng (芦笙) and etc.

For the future work, two issues can be pursued.

- (1) In the training, the sampled music piece is divided into multiple segments, each has 10 sec in length. While this is simple and effective, it may miss the long term correlation in a music piece. The use of longer segments (e.g., 30 sec or longer) may be beneficial for slow mode music, though, the training will be longer.

(2) The emotional features of the music pieces can be extracted, classified and labeled in the training process to further optimize the generated music.

Declarations

Data availability

All the data that support the findings of this study as well as the code for data preprocessing and analysis are available on request from the corresponding author.

Acknowledgement

We wish to thank for Prof. Wu and Miss Yang of Xinghai Conservatory of Music for their suggestions.

Author contributions

S.C. collected data, analyzed algorithm and conducted RL modelling, and wrote the first draft. Y.Z. and R.X. supervised the study, advised on algorithm analyses, and revised the paper. All authors interpreted results and approved the final version for submission.

Funding

We acknowledge the financial support of the Natural Science Foundation of Guangdong Province (Grant No. 2020A1515010621), GDNRC[2020]031, and the Strategic Priority Research Program of the Chinese Academy of Sciences (class A) (Grant No.XDA22040203).

Competing interests

The authors declare no competing interests.

Additional information Supplementary Information

Correspondence and requests for materials should be addressed to S.C.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Wang, P.K. Study on the Composition Technique of Chen Zhe Guzheng Works– Take five works for example. (Henan university Press, 2020)
2. Li, B. The historical inheritance and artistic charm of guzheng are briefly introduced. Song of the yellow river. 4, 9 (2020).
3. Yuan. J.F. National Instrumental Music (Revised Edition). (Beijing: Higher Education Press, 2004)

4. Li. M. Basic Course of Guzheng. Beijing: International Culture Publishing Company. 1, 52 (2020).
5. Westergaard P. *et al.* Experimental Music. Composition with an Electronic Computer. *Journal of Music Theory*. 3, 302 (1959).
6. Allan M. Harmonizing Chorales by Probabilistic Inference. *Proceedings of Advances in Neural Information Processing Systems*. 17 (2005).
7. Simon I. *et al.* MySong: automatic accompaniment generation for vocal melodies. *Conference on Human Factors in Computing Systems*. (2008).
8. Andries V. *et al.* Music Generation with Markov Models. *IEEE Multimedia*, 18, 78–85 (2011).
9. Franklin JA. Recurrent Neural Networks for Music Computation. *Inform's journal on computing*, 18, 321–38 (2006).
10. Hadjeres G. *et al.* DeepBach: a Steerable Model for Bach Chorales Generation. *34th International Conference on Machine Learning*, 3, 2187 – 2196 (2016)
11. Schuster M.P. *et al.* Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45, 2673–2681 (1997).
12. Schmidhuber J. *et al.* A First Look at Music Composition using LSTM Recurrent Neural Networks. *Idisia Usi Supsi Instituto Dalle Molle*, (2002).
13. Chung J. *et al.* Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint <https://arxiv.org/abs/1412.3555>* (2014)
14. Lambert AJ. *et al.* Perceiving and predicting expressive rhythm with recurrent neural networks. *Proceedings of the 12th International Conference in Sound and Music Computing*, 265-272 (2015).
15. Choi K. *et al.* Convolutional Recurrent Neural Networks for Music Classification. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2392 – 2396 (2017).
16. Kang S I. *et al.* Improvement of Speech/Music Classification for 3GPP EVS Based on LSTM. *Symmetry*, 10, 8 (2018).
17. Tang CP. *et al.* Music genre classification using a hierarchical long short-term memory (LSTM) model. *Proceedings of SPIE - The International Society for Optical Engineering*, 10828, 108281B - 108281B-7 (2018).
18. Tatar K. *et al.* Latent Timbre Synthesis: Audio-based variational auto-encoders for music composition and sound design applications. *Neural computing & applications*, 33, 67–84 (2020).
19. Goodfellow I. *et al.* Generative adversarial nets. *Advances in neural information processing systems*, 27: 2672–2680 (2014).
20. Guan F. *et al.* A GAN model with Self-attention Mechanism to Generate Multi-instruments Symbolic Music. *International Joint Conference on Neural Networks*, 1-6 (2019).
21. Li S. *et al.* INCO-GAN: Variable-Length Music Generation Method Based on Inception Model-Based Conditional GAN. *Mathematics (Basel)*, 9 (2021).
22. Meng Z. Research on timbre classification based on BP neural network and MFCC. *Journal of Physics Conference Series*, 1856, 012006 (2021).

23. Zhu D. Research on Inheritance and Innovation of Chinese Guzheng Artistic Performance Techniques. (Hebei Normal University Press, 2011).
24. Sutton R. *et al.* Reinforcement learning: An introduction (Adaptive computation and machine learning). IEEE transactions on neural networks, (1998).
25. Ye H. *et al.* Deep reinforcement learning based resource allocation for V2V communications. IEEE Transactions on Vehicular Technology, 68.4, 3163–3173 (2019).

Tables

Tables 2 & 3 are available in the Supplementary Files section.

Figures

Figure 1

Guzheng, a traditional Chinese music instrument

Figure 2

The sound waves of different music instruments when playing a same piece of song with the same volume.

Figure 3

Music format conversion

Figure 4

The similarity matrix of a sample music

Figure 5

An example of melody extraction, the melody is marked by yellow

Figure 6

A LSTM unit

Figure 7

Model demonstration

Figure 8

The process of agent-environment interaction in RL

Figure 11

A sample music generated by the LSTM

Figure 12

A sample music generated by LSTM with extracted melodies

Figure 13

The relationship between the loss and the average rate of return of RL

Figure 14

A sample music generated by the RL

Figure 15

Generative music process

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Table23.docx](#)